DOCUMENT RESUME

ED 291 252 FL 017 180

AUTHOR Rosenfeld, Samuel A.; Sable, Jerome
TITLE Requirements for LINCS File Management System. LINCS Project Document Series. LINCS #8-69.
INSTITUTION Center for Applied Linguistics, Washington, D.C.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE Jun 69
GRANT GN-771
NOTE 45p.; Best copy available. Some pages may not reproduce well.
AVAILABLE FROM Clearinghouse for Federal Scientific and Technical Information, U.S. Department of Commerce, Springfield, VA 22151 (Order No. PB 186 472; $3.00 hardcopy, $2.65 microfiche).
PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Computer Oriented Programs; *Databases; *Data Processing; *Information Needs; Information Networks; Information Science; *Information Systems; *Linguistics
IDENTIFIERS *Language Information Network Clearinghouse System

ABSTRACT

The report discusses the file management requirements of a computer-based information storage and retrieval system for the Language Information Network and Clearinghouse System (LINCS). It discusses a hypothetical structure for the LINCS system, requirements of file management, long-range trends in data management technology of interest to the LINCS problem, and evaluation criteria applicable to LINCS. (MSE)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

ED291252

CENTER FOR APPLIED LINGUISTICS

LANGUAGE INFORMATION NETWORK AND CLEARINGHOUSE SYSTEM (LINCS)

# REQUIREMENTS FOR LINCS FILE MANAGEMENT SYSTEM

by

Samuel A. Rosenfeld and Jerome Sable

Auerbach Corporation

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

GRT

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

LINCS PROJECT DOCUMENT SERIES / NATIONAL SCIENCE FOUNDATION GRANT

LINCS #8-69 June 1969 NSF GN-771

CENTER FOR APPLIED LINGUISTICS, 1717 MASSACHUSETTS AVENUE, N.W., WASHINGTON, D.C. 20036

FL017180



BEST COPY AVAILABLE

Copies of this document may be ordered from the Clearinghouse for Federal Scientific and Technical Information, U. S. Department of Commerce, Springfield, Virginia 22151. The order number is PB 186 472. The price is $3.00 for hard copy, and $0.65 for microfiche.

TECHNICAL REPORT
1604-100-TR-1

# REQUIREMENTS FOR LINCS FILE MANAGEMENT SYSTEM

SUBMITTED TO

CENTER FOR APPLIED LINGUISTICS
ATTN: MR. A. PIETRZYK

30 JUNE 1969

AUERBACH Corporation
121 N. Broad Street
Philadelphia, Pennsylvania 19107

TABLE OF CONTENTS


| Paragraph | Title | Page |
|---|---|---|
| 1. | INTRODUCTION | 1-1 |
| 1.1 | SCOPE | 1-1 |
| 1.2 | BACKGROUND | 1-4 |
| 2. | A STRUCTURAL DESCRIPTION FOR LINCS | 2-1 |
| 2.1 | TYPES OF BIBLIOGRAPHIC DATA ELEMENTS | 2-2 |
| 2.1.1 | Possible Combinations of Bibliographic Data Elements | 2-3 |
| 2.2 | LEVELS OF ACCESSIBILITY | 2-4 |
| 2.3 | MODES OF USER INTERACTION WITH THE LINCS SYSTEM | 2-7 |
| 2.3.1 | Combined | 2-8 |
| 2.3.2 | Separate | 2-8 |
| 2.3.3 | Remote/On-Line | 2-9 |
| 2.3.4 | Remote/Off-Line | 2-9 |
| 2.3.5 | Integrated | 2-10 |
| 3. | REQUIREMENTS FOR GENERALIZED FILE MANAGEMENT IN LINCS | 3-1 |
| 4. | LONG RANGE TRENDS IN DATA MANAGEMENT | 4-1 |
| 5. | EVALUATION CRITERIA FOR GENERALIZED FILE MANAGEMENT SYSTEMS | 5-1 |
| 5.1 | DESIGN OBJECTIVES | 5-2 |
| 5.1.1 | Responsiveness | 5-2 |
| 5.1.2 | Operations | 5-5 |
| 5.1.3 | Structures | 5-7 |
| 5.1.4 | Language Elements | 5-9 |
| 5.2 | APPLYING EVALUATION CRITERIA | 5-10 |
| 5.2.1 | Why Evaluation is a Problem | 5-10 |
| 5.2.2 | How Criteria Have Been Applied | 5-11 |
| 6. | BIBLIOGRAPHY | 6-1 |

# 1. INTRODUCTION

## 1.1 SCOPE

This report discusses the requirements of a generalized file management system for LINCS (Language Information Network and Clearinghouse System).

The flow of language information with books, journals, conferences, and reports is already inundating us. And it is getting worse every day. Based on realistic near-term projections of the Center for Applied Linguistics, the language community will be faced with the accession of almost 20,000 bibliographic items per year. This number will almost certainly continue to grow. Although no single individual is expected to remain current with all the items, the task of sorting out which items are indeed relevant to his interests becomes increasingly onerous.

Fortunately, concurrent with the increasing flood of language information we have been developing new techniques for coping with masses of information. The increased speed, decreased cost, increased computational power, and decreased

space requirements of the modern digital computer are finally beginning to realize the grandiose promises of earlier years. But most pertinently, new software methods are emerging that will be powerful aids in processing the information and data. In particular, large amounts of data which reside in files, like these data to be stored for the language community, can now be processed much more conveniently and efficiently. A data processing system need not be a Procrustean bed for the language information user. In fact, it is primarily for the convenience and efficiency of the user that a system should be designed.

File management is the center of the data processing system that will serve the users. File management simply refers to the functions of storage, retrieval, updating, and querying of data. Clearly, the success of a computer-based information system for language will depend greatly on the quality of the file management system to be chosen. This report will consider the characteristics of software packages, both in existence and projected for the near future, that will satisfy the file management requirements of LINCS. The cost of software has frequently been underestimated. For example, in one large public project the software was estimated to cost $4 million. It eventually cost $26 million. While this example may be rather isolated, it does suggest the need for a realistic appraisal of the implications of system design decisions on the file management software package.

The LINCS file management system must support the storage and retrieval of bibliographic references, subject terms, abstracts, and full text of technical documentation in the field of linguistic science. Furthermore, a file management

system will be required for retrieval of formatted linguistic data, and for the direction of LINCS itself, with respect to costs, user statistics, etc. A generalized file management system is a computer-based system that separates the language information user from the mechanics of data processing. A generalized file management system may be considered for LINCS because it offers flexibility in input and output formats and search strategies, and because it adapts to changing requirements in a cost-effective manner.

The term data management is used here in its most generic sense and encompasses all automatic processing of data. As such, it includes the fields of generalized data management, file management systems, information storage and retrieval systems, and specific application-oriented data processing systems. The areas of technology in data management that apply specifically to LINCS fall into two categories:

(1) Generalized data management systems that accept the definition of complex data structures of a formatted nature, prepare directories and definition tables of those data, and provide access to those data for users and programmers; and

(2) Information storage and retrieval systems that provide special mechanisms for aligning the vocabulary of untutored users against the vocabulary of the indexers and authors of the documents in a reference-providing system.

This report concentrates on the file management system requirements for LINCS and develops the evaluation criteria applicable to generalized file management systems pertinent to the LINCS problem. The report discusses a hypothetical structural description of a LINCS system, the requirements for the LINCS generalized file management, the long-range trends in data management technology of interest to the LINCS problem, and the evaluation criteria applicable to a LINCS file management system.

## 1.2 BACKGROUND

The Center of Applied Linguistics (CAL) has initiated a program to develop an information system for the language sciences. This program, described in the CAL proposal to the National Science Foundation,* involves a number of concurrent efforts on the part of CAL and its subcontractors. The project, now in the system design phase, encompasses two concurrent studies:

(A) Advanced studies toward an optimal system configuration, and

(B) Studies of priority components for the system.

The second effort, in turn, involves two sub-areas:

(B1) Indexing Tool Development, and

(B2) System Automation Studies.

This report covers Project B2, System Automation Studies, File Management System. Further background is contained in the above referenced proposal.

---

* - CAL Proposal, "An Information System Program for the Language Sciences, Stage 2: System Design."

## 2. A STRUCTURAL DESCRIPTION FOR LINCS

LINCS will be an adaptive network of individuals and organizations that will provide products and services to facilitate the transfer of linguistic information, in a variety of media, among the scientific community. The network will presumably include one or more central clearinghouses with computer-based facilities and a variety of terminal nodes consisting of user organizations, individuals, and other existing specialized information centers.

The following elements must be defined in the structural description of LINCS:

(1) Types of data to be handled and their structural organization within the system,

(2) Levels of accessibility to data provided through various automatic storage facilities,

(3) Ways in which individuals and other automated data centers will interact with LINCS data and network,

(4) Function of automatic data processing hardware and software.

These items are discussed in the subsequent paragraphs of Section 2 of this report. In addition, Section 2 identifies potential alternate modes of system interaction between LINCS and external bibliographic data bases.

The alternative modes of system interaction are defined according to the following characteristics:

(1) Bibliographic data elements to be input, stored, and output,

(2) Modes of user query,

(3) Modes of responses to LIMCS clients.

## 2.1 TYPES OF BIBLIOGRAPHIC DATA ELEMENTS

A library or bibliographic information system deals with information about documents (i.e., document surrogates), with information contained in documents, and with documents themselves. Therefore, the basic inputs and outputs to LIMCS will be various combinations of bibliographic data elements and documents. The possible inputs and outputs that may be utilized in an interactive network are indicated below:

(1) Abstracts or annotations - these may be either informative or indicative in kind, and may be assigned by an author or by a cooperating system.

(2) Full-text documents - these may be either the original published form, a reproduced hard copy, microform, or some other nonpaper representation.

(3) Bibliographic citations - these commonly include (as appropriate) author, article title, journal title, volume, issue number, pagination, date, imprint, report number, and source of document copy. It is also possible to cite the place of publication of a document surrogate in a secondary source, in addition to giving the primary citations.

(4) Accession or call numbers - these are secondary notations used to identify a document (or document surrogate) in a particular information system. Under certain circumstances such a number may be considered as part of a citation or as a substitute for the citation.

(5) Indexing information - this commonly includes subject terms and classification numbers, and may also involve personal and corporate author names.

2.1.1 Possible Combinations of Bibliographic Data Elements

Various combinations of the previously defined bibliographic data elements could be used for inputting information from other data bases and in responding to the queries of LINCS clientele. The relationships between the data received and held by LINCS and the responses to its users may now be defined.

Table 1 displays and defines relationships between the possible types of LINCS responses to its clientele and the possible data element combinations extracted and/or converted from other bibliographic data bases and held by the system. Ten relationships have been identified. These relationships are indicated in column 1 by Roman Numerals. Column 2 lists, for each relationship type, a possible combination of data elements suitable for response by LINCS to clientele queries. There are four basic types of responses. The third column lists, for each relationship type, a combination of data elements extracted from other data bases which may be held by LINCS. The fourth column lists those data elements that must be input processed in some manner to make them compatible with LINCS holdings. The last column describes briefly the types of input processing required for each relationship type.

Possible data element responses appear in column 2. Possible combinations of data element holdings appear in column 3, and possible combinations of data element input processing appear in columns 4 and 5. The ten relationship types (type I to type X) are developed because of different possible relationships between the combinations, appearing in the various other columns, which constitute responses, holdings, and input processing. The basic variables are the responses and the holdings; all other aspects of each relationship type are derived therefrom.

Table 1 thus sets forth the relationships between the possible responses to clientele (as a result of queries) and the possible data element combinations extracted by or converted by LINCS from other bibliographic data bases. The ten relationship types set forth in Table 1 provide one basis for determining all possible alternatives for inter-system interaction.

Table 1 deals with seven different types of bibliographic data elements.

(1) Annotations or abstracts

(2) Full-text document items

(3) Accession or call numbers

(4) Citations, not including accession or call numbers

(5) Indexing information as assigned by the source data base, including subject, author, and publishing source terms

(6) Common or converted indexing information capable of being merged with LINCS' own self-originated data base and searchable within LINCS' software and hardware

(7) Sources other than LINCS from which full-text documents should be obtained

## 2.2 LEVELS OF ACCESSIBILITY

Any automatic data processing system contains one or more central processors and a number of various storage devices. Storage devices differ with respect to the cost of information storage, storage capacity, access time and mode of access. When a large volume of information must be stored, the economic effectiveness of the various levels of storage devices must be considered. An important aspect of the system design will be to specify the kind of information and the amount of storage required at each level of accessibility. The

TABLE 1. RELATIONSHIPS BETWEEN POSSIBLE TYPES OF LINCS RESPONSES TO CLIENTELE AND POSSIBLE DATA ELEMENT COMBINATIONS HELD BY LINCS

| Type | Data Element Responses to Clientele Queries | Data Element Combinations Held. | Data Elements To Be Input Processed | Input Processing Requirements |
|---|---|---|---|---|
| I | Annotations or abstracts with citations and call or accession numbers. | Annotations or abstracts with citations and call or accession numbers, full-text documents, and indexing information as assigned by source. | Same as data element combinations held by the system. | Copying, reproduction, character code translation, record reformatting (as required by physical medium, character code, and format of input and by selected mode of query placement). |
| II | Same response as for relationship type I. | Same elements held as for relationship type I, but indexing information is converted to be merged with master data base. | Same input processing as for relationship type I, except for conversion of indexing information. | Same processing as for type I, except for conversion of indexing information to permit merging with master data base. |
| III | Same response as for relationship types I & II. | Same elements held as for relationship type I, but full-text documents are not held. | Same as data element combinations held by the system. | Same as for relationship type I, except that reproduction of full-text documents need not be considered. |
| IV | Same response elements as for relationship type III. | Same elements held as for relationship type II, but full-text documents are not held. | Same elements to be processed as for relationship type II, except that full-text documents are not processed. | Same as for relationship type II, but reproduction of full-text documents need not be considered. |
| V | Citations and call or accession numbers. | Same elements held as for relationship type I, but annotations or abstracts are not held. | Same as data element combinations held by the system. | Same as for relationship type I, except that annotations or abstracts need not be processed. |

2-5

TABLE 1. RELATIONSHIPS BETWEEN POSSIBLE TYPES OF LINCS RESPONSES TO CLIENTELE AND POSSIBLE DATA ELEMENT COMBINATIONS HELD BY LINCS (continued)

| Type | Data Element Responses to Clientele Queries | Data Element Combinations Held | Data Elements To Be Input Processed | Input Processing Requirements |
|---|---|---|---|---|
| VI | Same response as for relationship type V. | Same elements held as for relationship type II, but annotations or abstracts are not held. | Same as for relationship type V, except for conversion of indexing information. | Same as for relationship type V, except for conversion of indexing information to permit merging with master data base. |
| VII | Same response as for relationships V & VI, plus sources where full-text documents can be obtained. | Same as for relationship type V, except full-text documents are not held. | Same as data element combinations held by system. | Same as for relationship V, except that reproduction of full-text documents need not be considered. |
| VIII | Same response as for relationship type VII. | Same as for relationship type VI, except full-text documents are not held. | Same as for relationship type VII, except for conversion of indexing information. | Same as for relationship type VII, except for conversion of indexing information to permit merging with master data base. |
| IX | Same as for relationship types I, II, & III. | System holds only the indexing information as assigned by source. | Same as for data element combinations held by the system. | Copying, character code translation, record reformatting, and merging (with notation of source) to provide a combined vocabulary list. |
| X | Same response as for relationship types VII & VIII. | System holds only the indexing information as assigned by source, (type IX). | Same as for data element combinations held by the system, (type IX). | Same as for relationship type IX. |

storage devices range over the following levels:

(1) Immediate random access, magnetic core memory, where information is transformed, manipulated, and executed in the central processor,

(2) A random access backing store for information frequently needed, and requiring rapid accessibility, such as the system directory and indexes to information at lower levels of accessibility,

(3) A random access device with a slower access speed which may be used to store the document records containing basic bibliographic information such as accession number, author, title, and subjects,

(4) Lower level storage media such as magnetic tapes which may be used to store the document surrogates or the documents themselves, containing such things as the abstract or the full text of the document.

In addition to the bibliographic data elements, the data base contains information required for the operation of the center itself; the programs and active routines that respond to user inquiry and implement the various search techniques. This category includes routines that perform the cost-accounting and effectiveness evaluation of the system, thus permitting designers and administrators to monitor system performance and obtain insights into possible performance criteria and system improvements.

## 2.3 MODES OF USER INTERACTION WITH THE LINUS SYSTEM

Initial inspection has identified five basic modes by which data contained in, or extracted from, external data bases could be queried in order to satisfy the needs of LINUS clientele:

(1) Combined

(2) Separate

(3) Remote/On-Line

(4) Remote/Off-Line

(5) Integrated

These five modes of query placement, when used in combination with the previously defined types of relationships between LINCS responses to clientele and LINCS data element holdings, provide the means of defining basic alternatives of system interactions

2.3.1 Combined

In the combined mode of query placement, the system must convert extracted portions of other data bases to a common file structure (separate from LINCS own data base), to a common computer medium, and to a common format, so that all the extracted portions can be searched with LINCS software and hardware but by the use of different terminologies. The advantages of such an approach include: (1) LINCS control of response time, and (2) ability to search all areas of the extracted data base in one operation with one set of software. The problems of this approach include: (1) difficulty of determining the current relevant portions of external data bases, (2) the on-going maintenance of several conversion procedures - procedures which are dictated by decisions not under LINCS control, (3) finding personnel to handle the many search terminologies, and (4) inefficiency of writing many search formulas for a single request.

2.3.2 Separate

In the separate mode of query placement, the system converts the extracted portions of other data bases to a common computer medium at a single location, so that each data base can be searched only by using its appropriate software and indexing terminology. The advantages of such an approach include: (1) less initial effort required than in the combined and integrated modes to establish an operational center, and (2) LINCS control of response time. The

problems of this approach include: (1) the difficulty of determining the current relevant portions of external data bases, (2) the on-going maintenance of several different software terminology and file structure packages which are subject to change beyond LINCS control, (3) finding personnel to handle the many query languages and search approaches, (4) the inefficiency of writing many search formulae for a single request, and (5) determining a reasonable search priority of data bases to obtain maximum retrieval efficiency.

2.3.3 Remote/On-Line

In this mode of query placement, the system searches an external data base via remote, on-line terminals, using the external system's index terminology, software, and hardware (except for terminals, etc.). The advantages of this approach are that LINCS has no maintenance tasks, and the entire data base may be searched without predetermining relevant areas. On the other hand, LINCS has little control over the system. Difficulties arise in finding search formulators capable of handling the variety of techniques and languages required to search many data bases. Experience with the Neurological Information Network has shown that using two or more vocabularies covering similar material results in negative interference so that one person cannot efficiently handle more than one search strategy. Search prioritization among data bases would be difficult to establish. In addition, on-line search capabilities for bibliographic information have not been completely refined and will probably not be realizeable for several years.

2.3.4 Remote/Off-Line

The remote off-line mode is similar to the remote on-line mode; however queries are placed by mail, TWX, telephone, etc. One significant advantage of

this approach is that each data base would be searched by personnel familiar with the system, which would have a positive effect on the output. In addition, the entire data base could be searched without prior determination of relevant portions. The approach also frees LINCS of maintenance responsibilities. The problems of this approach are that LINCS has no quality control and no control over response time. A further problem arises in determining which services to interrogate.

2.3.5 Integrated

In the integrated mode of query placement, LINCS converts extracted portions of other data bases to its own file structure, computer medium, format, and indexing terminology, thereby permitting the merging of such data into LINCS' own master bibliographic data base for searching and/or announcement. The approach has many advantages, including:

(1) Relative ease of system maintenance through use of tables, which permits acceptance of diverse inputs and production of diverse outputs. In addition, changes in external data bases over which LINCS has no control could be accommodated by changing the tables.

(2) A single query language. Writing a comprehensive search that will extract only relevant data requires a great deal of experience with both the vocabulary and the data base. The use of a single language and data base simplifies the problem.

(3) Control of response time.

(4) Ability to search all areas of the data base in one operation to retrieve maximal relevant information.

While the advantages of this approach are great, they may well be outweighed by the problems. Determining what constitutes the relevant portion of an external data base is a difficult task, particularly when that portion is to be used for demand searches. Designing a machine system to process many

diverse formats, yet still provide flexibility and ease of maintenance, is also a complex task, which adds a large initial cost to the system. The most difficult task in systems of this type (as evidenced by AUERBACH's experience with the Neurological Information Network of the NINDB) is to resolve incompatibilities of thesaurus and indexing approaches. Developing a single query language and technique that will apply to a consolidated data base requires a substantial initial intellectual effort.

## 3. REQUIREMENTS FOR GENERALIZED FILE MANAGEMENT IN LINCS

The costs of system design, programming, and program maintenance have historically been a very large part of the costs of developing and running a data processing or information center. In an effort to reduce this cost, various approaches have been implemented for generalizing the data processing functions involved in a data center. This generalization of function such as data input, storage, retrieval, data file maintenance, and reporting is collectively known as a generalized data management system approach. As a rule, systems are defined imprecisely at the outset and must undergo modification to meet user needs effectively. The true nature of the data and the needs of the users are determined only through experience with the system. Furthermore, the needs of the users change with time so that repetitive adaptation is necessary.

The LINCS file management system should simplify the development and modification of programs and expedite the solution of users' problems. The achievement of this goal will reduce the cost of construction, modifying, executing programs and solving users' problems. To achieve this, the software system should

introduce the ability to improve the LINCS file management flexibility by providing an optimum approach to the following LINCS objectives:

(1) Input from a variety of sources including both local keyboarding and machine-readable records created by other organizations,

(2) Output to produce printer primary and secondary publications with optional indexes,

(3) Ability for dissemination of machine-readable records to other publication and information centers,

(4) Storage and/or output of managerial control data,

(5) Permanent storage of the data for possible later users in a retrieval and dissemination system.

## 4. LONG RANGE TRENDS IN DATA MANAGEMENT

The software associated with every computing system can be broken down into a number of distinct levels with respect to the distance between the module in question and the computing hardware itself. Effective utilization of the computer in any information system depends on having a number of modules that can be identified as the operating system for the computer. The operating system consists of a number of distinct modules associated with the management of activities in the computer, the allocation of computing resources to jobs or users of the computer, and the sequencing of various jobs and tasks. Another important dynamic function of the operating system is the storage and retrieval of data on the various levels of storage devices associated with the computing system.

These functions of resource allocation, activity management, and management of data on the storage devices constitute the foundational elements of an operating system. They are foundational because they dynamically interact with the running program that makes use of their services and because they are resident in the computing system's memory and therefore, can be called upon to service other functional elements which are more application oriented or less dynamic in their interaction with the system.

In addition to these foundational elements, a number of other functional modules are included in the operating system. These can be viewed as super-structural elements. One class of super-structural elements are the dynamic elements associated with interaction with the user such as the console monitor or batch supervisor, if it is a non-interactive system. Another super-structural category includes the program development tools and services. These include the assembler language, translators, the compilers, the linkage editor and the loader. A third category of the super-structural elements can be termed the system support jobs. These are routines that help the user use the system, but are jobs or tasks in the same sense that the users' programs are jobs or tasks for the system. Routines such as library maintenance routines, data definition routines, job definition routines can all be categorized as system support functions in the operating system.

In addition to these elements of the operating system, which in some sense simplify the users' interface with and utilization of the hardware, there are a number of generalized jobs that are not unique to one particular application. This latter group of elements includes some current software systems known as data management or file management systems, report writers, query system, information retrieval systems, and document processing systems. Because typically these systems are not fully integrated with the operating system, each presents its own peculiar input language, user language, and constraints of operation.

One such software package called Document Processing System (DPS of IBM) would seem at first glance to be very appropriate to the LINCS type of application. DPS is oriented to documentation systems that provide references. It accepts the definition of the format of a record which represents the reference-providing information for a single document. The record can be indexed and queried by the subject matter so that the user can receive the list of document references, document numbers or even abstracts of the documents. Yet, DPS presents a serious contraint to the user. The files of DPS must exist on-line in a single volume on

a disc storage device. This limits its effective use to relatively small data collections.

Several large-scale generalized data management systems have been under development for some time without fully realizing the initial goals set forth by their designers. Systems such as IBM's GIS, AUERBACH's DM-1, IBM's Information Management System (or IMS), and MITRE's ADAM can be included in this category. The shortcomings of these systems and the cost of their development can be attributed in part to the difficulty of integrating them with existing operating systems supplied by the manufacturers. Operating systems such as OS 360 for the IBM 360 series of computers are very complex; they represent the high overhead in the mounting storage that the user must devote to operating the system functions; and they are difficult for users to modify. Indeed, user modification is generally impractical because manufacturers frequently change these operating systems, making it difficult to maintain any one special version.

The long-range trend in the development of operating systems and data management systems is toward the gradual integration and sophistication of the data management functions which are integrated with the foundational elements of the operating system. Structurally, the data management systems of the near future will exhibit a distinct hierarchy of functions. Close to the foundation of the operating system will be such functions as those that retrieve and store fixed blocks of data from the secondary storage devices and move them into and out of the computer system's primary memory. More sophisticated levels of data service support to the programmer will be built on these machine-oriented functions. These will provide the ability to handle variable length streams of data, to build and manipulate linked data structures, and to define and manipulate files of information to be stored on secondary storage devices.

The file management system for LINCS exhibits characteristics of this last category of data management. In LINCS, the more primitive types of data management functions will be built into the operating system and will provide the ability to define the files and indexes and retrieval strategies suitable for very large files of reference-providing information. This type of data management system will be characterized by the ability to define a structured vocabulary called a thesaurus that will align the vocabulary of the untutored user to the rigid terminology of the indexers. The user will be able to conduct a multi-stage dialogue with the system during which he will learn the vocabulary representing the areas of his interest. Once having learned the proper index terms and their generic/specific relationships and perhaps having learned which terms are synonymous with others, he will be able to formulate a meaningful inquiry to retrieve the desired information in a discriminatory way.

Another trend of future data management systems is the gradual standardization of the language used to describe the data structures in these systems. This category of language is known as data description languages or DDL's. Using a DDL, a system designer might describe the terminology and structure of a file he wishes to define to a system, or to transmit from one system to another. The use of a DDL and an appropriate interpretor will enable the designer to create appropriate directories and to transmit information from one center to another in an intelligible manner.

One important trend already apparent in the field of scientific documentation is the creation of a number of specialized information services and centers and the attempt to create mechanisms that permit their interaction as a network of information services. The use of data communication lines between centers and the adoption of uniform terminology and thesauri support this trend.

AUERBACH is currently engaged in a project to define the capabilities of the National Agricultural Library for the Department of Agriculture. It is anticipated that the National Agricultural Library will be one such center in a network of information centers, consisting primarily of the National Library of Medicine, The Library of Congress, and The National Agricultural Library and their adjuncts. Other candidate information centers for inclusion in this network are the Biosciences Information Services of Biological Abstracts, the Library of the United States Atomic Energy Commission, the Clearinghouse for Federal Scientific and Technical Information, the Library of Congress, the National Library of Medicine and MEDLARS System, the Institute for Scientific Information, and the Chemical Abstracts Service. Logically, the LINCS system should be one such information center in a national network.

Much of the research devoted to the problems of designing an information center and creating a network of such centers is certainly applicable to the LINCS problem.

## 5. EVALUATION CRITERIA FOR GENERALIZED FILE MANAGEMENT SYSTEMS

This section discusses the criteria for evaluating file management systems and techniques pertinent to LINCS. These criteria may be applied to file management systems recently developed or currently being developed.

AUERBACH and other agencies have already conducted several surveys and critical evaluations of data management systems. These are listed in the Bibliography included in this report. Some surveys have concentrated on tabulating a number of features and parameters which file management systems may or may not possess. Over 100 such parameters have been tabulated in reports by the Fry et al (see Bibliography, item 2) and the Codasyl Committee, item 1). Comprehensive listings of features such as those tabulated can present a rather bewildering array of factors to be evaluated. These factors must be placed in proper perspective in assessing their pertinence in the LINCS system. The difficulty in applying the parameters appearing in prior reports stems from the fact that each previous report addressed a problem slightly different from LINCS. In the Fry study[2], and in the Codasyl study, only existing generalized data management systems were considered. Furthermore, the studies did not consider combinations of file management systems that did not fall strictly into the category of a "generalized data base management system."

The features of a data management system that relate to the LINCS problem will be discussed in this section from several points of view. In effect, these features represent the criteria by which the systems can be studied and evaluated. These features are presented from four points of view.

(1) Design Objectives - Discusses the overall goals of the system, without regard to the various ways in which these goals may be realized.

(2) Operations - Discusses the various system functions and capabilities for accomplishing the design objectives.

(3) Structures - Discusses the system components (i.e., tables, data structures, and program module structures, their composition and interrelationships) used to perform system operations.

(4) Language Elements - Discusses the system commands and service calls which may provide an appropriate interface with the system users and programmers.

All of the software elements of the LINCS system can be evaluated from these points of view. These elements include the operating system with its machine-oriented job management and data management aspects, the programming language compilers, and the console monitors which allow user interaction with the system.

The File Management System (FMS) features are summarized in Table 2 and are discussed in the following paragraphs.

## 5.1 DESIGN OBJECTIVES

### 5.1.1 Responsiveness

The primary design objective of the FMS should be system responsiveness to user needs. User functions are discussed in Section 5.1.2.1. To the extent that the user deals directly with the FMS, it should be easy to use and learn. The FMS must provide quick response to service requests and rapid handling of search and update operations.

## DESIGN OBJECTIVES

### Responsiveness

- Ease of use
- Novice training
- Quick response, search, and update

### Adaptability

- Independence of logical data structure from JOSS
- Independence of logical data structure and program
- Ability to combine data in unforeseen ways
- Language and command definitional capability

### Efficiency in

- Block utilization
- Data representation for storage
- Indexing arrangements
- Retrieval strategy
- Updating methods
- Sharing of common data

### Reliability

- Control of authorized access
- Error recovery

## OPERATIONS

### User Functions

- Query
- Editing
- Updating
- Report generation
- Program entry
- Program execution
- Novice training

### Interface with

- User
- Program
- External file system
- Job management system

### System Functions

- Translation of data values (input, output, and storage)
- Data base updating
- Directory updating and indexing
- Data search
- Data search look-ahead
- Maintenance of data usage statistics
- User accountability
- Backup and failure recovery (job and data restart points)

## STRUCTURES

### General Considerations

- Data base structure and size
- Variable versus fixed block length
- Intra-block structure and format
- Data linkage
- Priority ordering of data segments in files

### System Structures

- Logical data directories
- Data file dictionary (item name dictionary)
- Indexes
- Access rights table

## LANGUAGE ELEMENTS

### User Languages for

- Program specification
- Program execution
- Data definition
- Data entry
- Query
- Output format specification

### Programmer Languages for

- Data updating
- Data retrieval
- Report generation
- Task calling
- Control transfers

STRUCTURES (cont.)

Item Structure

- Logical subdivision and relations among user items
- Degree of nesting permitted

System Program Modules

- Modularity
- Standard program interfaces
- Generality of program functions

LANGUAGE ELEMENTS (cont.)

System Languages

- Data coding schemes
- Input/output formats
- Interface with external file system
- Interface with JMS (Job Management Sys

5.1.1.2 Adaptability. The FMS must be capable of adapting to a wide variety of user needs and environmental changes. In order to extend the useful life of various parts of the system and to minimize the implications of changes, the logical structure of data should be kept independent of both the FMS and the using programs. The system should also be able to combine and use data in unforeseen ways, so that the data structures and organization do not rigidly determine the ways in which data may be used. Finally, the user should be allowed to define his own languages and commands to the system to accommodate special needs.

5.1.1.3 Efficiency. If the FMS is to meet effectively all the demands placed on it, operating efficiency is an important factor. To make maximum use of the available storage, methods of representing data for storage and methods of utilizing the space within data blocks should be carefully considered. Data of interest to more than one user should be capable of being shared, with proper attention paid to protecting the data and, where necessary, providing control over data access. Indexing arrangements are probably the crucial factor in determining the speed and flexibility of accessing data. Data retrieval strategy and the methods of updating and maintaining the data base will also play key roles in determing system efficiency.

5.1.2 Operations

5.1.2.1 User Functions. While the FMS user will ordinarily be an individual, task programs may also be considered users, inasmuch as a task program may call on services provided by the FMS. For maximum flexibility, both kinds of users should be able to call all FMS services, although the appropriate languages for doing so need not be the same.

Since querying or obtaining information from the system is the primary user function, the query facilities are extremely important. There should be a variety of ways of specifying conditions under which data are wanted. For the user who is not intimately familiar with the data base, it would be helpful to have a dialog query capability, in which the user would ask a series of increasingly specific questions, each based on the results of the previous ones, until the desired item was found. This dialog could also be the chief means of training novices in the use of the system.

Other user functions that may be called on by task programs as much as by human users include editing or arranging information for some specific purpose updating the data base, and generating reports.

5.1.2.2 Interfaces. The FMS occupies a rather central position in the system since it interfaces with human users, task programs, and the operating system. However, since the FMS isolates as well as connects these subsystems, changes to one subsystem should have a minimal effect on the others.

5.1.2.3 System Functions. The system functions are the built-in, intrinsic functions of the FMS, and bear a large part of the responsibility for achieving the FMS design objectives. First, there are the functions dealing directly with data, including translation of data values for input, output and storage, data base updating, and data retrieval. When data retrieval is sequential or patterned in some way, retrieval efficiency should be increased by performing a look-ahead operation in conjunction with the FMS. Next, there are supervisory data functions, including directory updating and data indexing (which should be done automatically whenever the data base is changed), and maintenance of data usage statistics. These statistics may be used to reorganize the data base in

a more efficient manner, either automatically or manually. Finally, the FMS should play some role in keeping track of each user's use of the system and should provide backup and failure recovery facilities through job and data restart points or other means.

5.1.3 Structures

5.1.3.1 General Considerations. Various general structural considerations affect the design of an FMS. The following structural features will be considered:

(1) The organization of data base;

(2) The expected size of the data base, and its implication on the system design;

(3) The length (fixed or variable) of the data block exchanged between the FMS and the operating system data management function;

(4) The logical structure and physical format of the data block;

(5) The ordering principle used to arrange data segments within files;

(6) The facilities which should be provided for data linkage.

5.1.3.2 System Structures. System structures are tables maintained by the system to describe and provide access to the data base. These structures include:

(1) Logical Data directories, which describe the logical structure of the data items and their logical position in the data base.

(2) Data file or item name dictionaries, which provide a cross-reference between the symbolic item names used by the external user and the structural or other codes used to identify the items internally.

(2) Indexes, which tell where in the data base certain data values may be found, thereby enabling the system to perform searches without accessing the data itself, or with minimal accessing. The type and amount of indexing are critical in determining the system effectiveness since search time can be minimized through adequate indexing, while excessively detailed indexing requires much time and space for maintaining the indexes.

(4) Access rights tables, which enable the FMS to keep track of which users are authorized to access each portion of the data base, and for what purposes.

Other system structures, such as a list of active tasks and their data requirements and a list of users and the extent to which they use system facilities, may also be required if the FMS is performing job management functions.

5.1.3.3 Item Structure. Item structure refers to the logical structure of individual data items. Items are divided into subitems, which may be further subdivided and ultimately divided into fields or values. Certain substructures may be repeated an arbitrary number of times; also certain substructures may be optional. Relations among items may be expressed implicitly by the logical nesting of items within other items, or explicitly by means of directories or various kinds of data linkage. The degree of complexity permitted in item structure is important. Allowing arbitrary complexity will entail a certain overhead in system development costs and running time, but may be justified because the system will be less subject to change arising from a need for data structures more complex than those originally envisioned.

5.1.3.4 System Program Modules. The organization of system programs comprising the FMS must also be considered. These programs should be as modular as possible in order to facilitate implementation, debugging, and documentation, and in

order to minimise the effects of changes. The use of a standard method for program interface will also contribute to these ends. Finally, program functions should be as general as possible so that the programs may lend themselves to uses not originally foreseen and thereby extend their useful life.

5.1.4 Language Elements

5.1.4.1 User Languages. Program specification languages are used to define task programs. A special language may be provided for this purpose, or the system may be built to accept the output of any standard procedural language processor. Program execution languages call for the execution of programs and supply them with necessary parameters. The user functions of defining data structures and entering data require appropriate languages. Language definitional facilities would be especially helpful for the data entry function, particularly where large quantities of data are involved. A query language is necessary to enable the user to retrieve information from the data base. The principal considerations here should be flexibility and the user's ability to obtain information in spite of a limited familiarity with the data base. The user must also be able to specify the form in which he wants the results presented. Hence, an output formatting or report generation language is required.

5.1.4.2 Programmer Languages. Programmer languages are those used by the task programmer to call on DMS services. The most important of these services are data updating, data retrieval, and report generation. In addition, a control language is needed so that tasks can call for the execution of other tasks and so that control may be passed from task to task and between tasks and the control system.

5.1.4.3 System Languages. System languages are those used by the FMS itself when it operates upon data and interfaces with other subsystems. Data coding schemes compress data in order to save space and also, possibly, to prevent unauthorized access. Also, data must be formatted appropriately for input/output operations. A primary interface of the FMS is with the operating system's data management routine. Symbolic block names and data blocks themselves are exchanged in both directions across this interface. The FMS also interfaces with the Job Management System (JMS) in the operating system; the FMS can use the JMS as an intermediary in dealing with the user.

## 5.2 APPLYING EVALUATION CRITERIA

### 5.2.1 Why Evaluation is a Problem

A number of evaluation criteria for FMS software have been discussed in the previous paragraphs. This in itself does little to solve the problem of determining the value of a given FMS in meeting the needs of the LHMCS. Neither does it determine the relative merit of competing software modules. Rather, it specifies what factors should be considered in evaluating a given system. No concrete procedure is known which can determine the value, or even relative merit, of a system. This is due to the existence of different kinds of evaluation criteria. The comparison of systems with different characteristics (such as one which performs only part of the required operations in an inflexible way to one which performs all of the operations required in a flexible way at much higher cost) must be accomplished in careful tradeoff studies by skilled analysts. Even after careful analysis, problems of this type may in fact have no definitive solution.

If careful analysis reveals that no algorithmic solution is feasible, one is free to look for effective heuristic methods - at least, for one which has the virtue of being readily applied. We now proceed to examine approaches to problems of this kind which have been used or proposed.

5.2.2 How Criteria Have Been Applied

5.2.2.1 Weighting. If one assumes that each of the criteria for evaluating a system is measurable, then, in general, a partial ordering is established among all systems being compared. Consider four systems, S, T, U, and V being evaluated under three critiera, A, B, and C, each criterion measured on a scale allowing a highest score of 10. The following result may be obtained:

| | S | T | U | V |
|---|---|---|---|---|
| A | 4 | 5 | 3 | 4 |
| B | 3 | 9 | 4 | 5 |
| C | 5 | 2 | 7 | 8 |

System V is uniformly better than System U for all criteria so that there is no difficulty in making a choice. System U can be eliminated from consideration. However, of the remaining systems, no one is uniformly better than any others and the best we can do is partially order the systems by criter. In order to break this impasse we could choose the system which rated highest in the largest number of criteria. System T would win by that measure, having the highest rating in criteria A and B. Another approach would be to choose the

system with the highest total point score. This approach leads to the selection of system V as shown below:

| | S | T | V |
|---|---|---|---|
| A | 4 | 5 | 4 |
| B | 3 | 9 | 5 |
| C | 5 | 2 | 8 |
| TOTALS | 12 | 16 | 17 |

However, if it were deemed that criterion B were (say) twice as important as criteria A or C, the scores with weighted valuedswould appear as follows:

| | S | T | V |
|---|---|---|---|
| A | 4 | 5 | 4 |
| B | 6 | 18 | 10 |
| C | 5 | 2 | 8 |
| TOTALS | 15 | 25 | 22 |

with system T winning. Choosing a weighting factor for each criterion in effect reduces the vector valued criterion to a scalar (single value) which assures the ability to transform the partial ordering into a total ordering. The most universally appropriate technique is to reduce each criterion to a dollar cost. However, even this measure is difficult to assign. For example, the "cost" of a given measure of reliability is the present value of the cost of the series of repairs that the system is expected to undergo (including cost of lost service).

But it is difficult to assess the (negative) "cost" of a system which has an outstanding adaptibility to changing requirements of data structures, or that presents an extremely well-engineered interface with its users.

5.2.2.2 Benchmark Problem. One approach to the evaluation of systems is through the use of benchmark problems. A benchmark problem is a complete simulation of a situation to which the system is expected to respond. The simulation may be designed to represent either a typical demand on the system, a situation of extreme stress on the system, or a scenario of samples from a mix of problems representing the projections for long-term demands on the system. Each of these types of benchmark problems, when used to gauge the performance of a system, either in terms of cost, responsiveness, or other factors, represents a particular bias but, depending on the system requirements, may be a valid gauge of system performance.,

For a system such as LINCS, which provides an information service to its subscribers which has an economic value (although perhaps an intangible one), a cost measure determined from a benchmark problem representing long-term demands seems to be most appropriate. Suitable cost components for user investments such as training and time at console should be considered, along with the cost of system purchase, operation, and maintenance. The system which satisfies the benchmark problem with the lowest overall cost is the one selected.

## 6. BIBLIOGRAPHY

(1) Codasyl Systems Committee: A Survey of Generalized Data Base Management Systems, May 1969 (Available through ACM).

The characteristics of the following systems are described in a common terminology:

| | |
|---|---|
| ADAM | (Mitre Corp.) |
| GIS | (IBM) |
| IDS | (GE) |
| ISL-1 | (Info System Leasing Corp.) |
| MARK IV | (Informatics) |
| MIPS/FFS | (IBM) |
| SC-1 | (Western Electric/AUERBACH) |
| TDMS | (SDC) |
| UL-1 | (RCA) |

A bibliography is included.

(2) Fry, J. P. et al: Data Management Systems Survey, Mitre Corp., January 1969.

This report presents the results of a survey of salient characteristics of a representative set of state-of-the-art data management systems. It is part of an effort to identify the state-of-the-art capabilities of data management systems for third-generation computer systems.

Section I of the report includes general descriptions of the systems surveyed and establishes the terminology for logical organization of data used in the survey.

Section II describes the capabilities surveyed and presents the survey results in tabular format.

The systems covered in this survey are:

| | |
|---|---|
| COGENT | (CSC) |
| DM-1 | (AUERBACH) |
| FORGE | (BURROUGHS) |
| GIS | (IBM) |
| IDS | (GE) |
| MANAGE | (SDS) |
| MARK IV | (INFORMATICS) |
| MIPS | (MMOSY IBM) |
| RAPID | (ARMY/CDC) |
| TDMS | (SDC) |

A bibliography is included.

(3) Landau, R.: Classified Bibliography on Bibliographic Data Base Interaction, Compatibility, and Standardization, AUERBACH 1530-100-TR, March 20, 1969.

(4) Pietrzyk, A. et al: File Management Techniques and Systems with Applications to Information Retrieval-- A Selective Bibliography, Center for Applied Linguistics, June 1968.

(5) Sable, J. D. and J. Cochrane: Date Management Systems Study, AUERBACH 1469-TR-4, April 1968.

(6) Ziehe, T. W.: Data Management: A Comparison of System Features, TRACOR 67-904-U, October 1967.